UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEMETRIUS MILTON MURATO

CLUSTERIZAÇÃO E ANÁLISE DE *TWEETS* COM FOCO EM POSTAGENS RELACIONADAS ÀS AÇÕES DA PETROBRAS

LONDRINA

2021

DEMETRIUS MILTON MURATO

CLUSTERIZAÇÃO E ANÁLISE DE *TWEETS* COM FOCO EM POSTAGENS RELACIONADAS ÀS AÇÕES DA PETROBRAS

CLUSTERIZATION AND ANALYSIS OF TWEETS FOCUSING ON POSTS RELETED TO PETOBRAS STOCKS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Produção da Universidade Tecnológica Federal do Paraná (UTFPR).
Orientador(a): Dr. Bruno Samways dos Santos.

LONDRINA

2021

DEMETRIUS MILTON MURATO

CLUSTERIZAÇÃO E ANÁLISE DE *TWEETS* COM FOCO EM POSTAGENS RELACIONADAS ÀS AÇÕES DA PETROBRAS

Trabalho de Conclusão de Curso de Graduação para obtenção do título de Bacharel em Engenharia de Produção da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 29/11/2021

Bruno Samways dos Santos
Professor Doutor
Universidade Tecnológica Federal do Paraná

Rafael Henrique Palma Lima
Professor Doutor
Universidade Tecnológica Federal do Paraná

Carlos Alberto Ribas
Professor Mestre
Universidade Tecnológica Federal do Paraná

AGRADECIMENTOS

Agradeço primeiramente a minha família, que sempre esteve ao meu lado durante toda a trajetória, possibilitando que eu tivesse essa vivência única.

Agradeço ao meu orientador, Bruno, que esteve presente em diversas atividades extracurriculares, e principalmente, por aceitar conduzir essa experiência.

Agradeço aos meus amigos e colegas, que estiveram envolvidos diretamente e indiretamente nessa jornada, me dando força para superar os obstáculos.

E também, a todos os professores da Universidade Tecnológica Federal do Paraná, pela excelência no ensino.

RESUMO

A Brasil, Bolsa e Balcão (B3), responsável por R$6,45 trilhões de reais movimentados no ano de 2020, contribui diretamente e indiretamente para o aumento das informações disseminadas pelas mídias sociais, impactando o mercado acionário. Por ser em grande quantidade, os investidores não conseguem analisá-las, então, ter um artifício que colabora para o agrupamento de notícias ligadas ao um mesmo assunto, pode contribuir para o desempenho dos investidores. Diante deste cenário, o presente trabalho utilizou o aprendizado de máquina não supervisionado para agrupar *posts* coletados do *Twitter* relacionados às ações da Petrobrás. Originando-se da coleta de dados por meio da sincronização com a plataforma Twitter API, foi realizado o pré-processamento baseado em técnicas de mineração de texto, aplicação de *Bag-of-Words* (BoW) e *Term Frequency-Inverse Document Frequency* (TF-IDF) para definir os termos mais recorrentes e o peso de cada *post* até a realização do agrupamento. Neste caso, para comparação, foi realizado um agrupamento direto da matriz obtida por TF-IDF e outro agrupamento após redimensionamento da matriz de pesos pelo *Principal Component Analysis* (PCA). Afim de confrontar e facilitar a visualização das principais diferenças, foram criados gráficos de dispersão e nuvens de palavras para cada agrupamento. Os resultados obtidos mostraram que realizar agrupamento em uma matriz redimensionada pelo *Principal Component Analysis* tem um melhor desempenho para a separação de textos relacionados entre si, contribuindo para a sua interpretação.

Palavras-chave: mineração de texto; análise de sentimentos; agrupamento; Petrobras.

ABSTRACT

Brasil, Bolsa e Balcão (B3), responsible for R$6.45 trillion in transactions in 2020, directly and indirectly contributes to the increase of information disseminated by social media, impacting the stock market. Because there is a large amount, investors cannot analyze them, so having an artifice that contributes to the grouping of news related to the same subject can contribute to the performance of investors. Given this scenario, the present work used unsupervised machine learning to group posts collected from Twitter related to Petrobras' stocks. Originating from data collection through synchronization with the Twitter API platform, pre-processing was performed based on text mining techniques, application of Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF) -IDF) to define the most recurrent terms and the weight of each post until grouping is carried out. In this case, for comparison, a direct grouping of the matrix obtained by TF-IDF and another grouping after resizing the weight matrix by the Main Component Analysis (PCA) was performed. In order to confront and facilitate the visualization of the main differences, scatter plots and word clouds were created for each grouping. The results obtained showed that performing grouping in a matrix resized by the Principal Component Analysis has a better performance for the separation of related texts, contributing to its interpretation.

Keywords: text mining; sentiment analysis; clustering; Petrobras.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

B3	Brasil, Bolsa, Balcão
BoW	*Bag of Words*
CEO	*Chief Enterprise Officer*
IP	*Internet Protocol Address*
JBS S.A	José Batista Sobrinho Sociedade Anônima
MPF	Ministério Público Federal
NLTK	*Natural Language Toolkit*
PCA	*Principal Component Analysis*
PCs	*Principal Components*
Re	*Regular Expression*
TF-IDF	*Term Frequency-Inverse Document Frequency*
WCSS	*Whthin Cluster Sum of Squares*

SUMÁRIO

1 INTRODUÇÃO

Atualmente a bolsa de valores brasileira, a Brasil, Bolsa e Balcão (B3) tem aproximadamente 3,3 milhões de contas (B3, 2020) que movimentaram um total de R$6,45 trilhões de reais no ano de 2020 segundo a *Money Times* (2021), por meio da negociação de ativos financeiros. Dentre esses ativos, as ações chamam a atenção pelo volume de capital movimentado e pela possibilidade de ganhos no curto e longo prazo. Investidores que têm como objetivo investir no longo prazo são conhecidos por fazerem parte da escola fundamentalista, ao buscar o valor intrínseco da empresa que depende de questões macroeconômicas, setoriais e de políticas internas da própria empresa. Por outro lado, existe a escola técnica que ao analisar gráficos verifica se o mercado está em tendência de alta ou baixa, e a partir desses resultados realizam compra ou venda de ações em um curto período.

Os investidores de curto prazo, conhecidos como *traders*, utilizam técnicas gráficas, volume de negociações e força do poder de compra e venda para a tomada de decisão. Com base nesses princípios de movimento de mercado, estudiosos estão utilizando o *machine learning* para tentar prever os preços do mercado de ações, e conforme aponta Raminelli e dos Santos (2019) as publicações de pesquisas relacionadas a *machine learning, data mining* no mercado de ações tem crescido significativamente nos últimos anos, Silva e Tessaro (2013) utilizaram a mineração de dados para analisar a correlação entre indicadores financeiros e variação de preços de ações, e Souza (2021) com o *software* Weka, aplicaram mineração de dados na previsão de preços de ações.

O *machine learning* trabalha com modelos matemáticos que utilizam dados estruturados do tipo quantitativos, e também não estruturados, como por exemplo textos. Ao trabalhar com dados estruturados, utilizam-se modelos que adotam parâmetros técnicos convertendo entradas de valores contínuos, como por exemplo o preço das ações, em entradas discretas que definem posteriormente uma saída, determinando se o mercado está em tendência de alta ou queda. De acordo com Galdi e Lopes (2008) a expectativa do mercado sobre o desempenho futuro de uma empresa e da economia contribuem para formação do preço de uma ação, porém o destaque é para o lucro, tal fator possibilita prever os preços das ações com base em dados comerciais. Em contrapartida, conforme aponta Seong e Nam (2021), o valor de uma ação pode ser avaliado através das notícias, uma vez que essa muda o

sentimento do investidor, além disso, existe facilidade em obter informações abundantes pela *internet*, tornando a informação essencial ao estimar os preços futuros das ações.

Sendo assim, o presente trabalho faz o uso de técnicas de aprendizado não supervisionado para agrupar e analisar tweets sobre as ações relacionadas à Petrobrás.

1.1 Objetivo Geral

Utilizar o aprendizado não supervisionado para agrupar *tweets* relacionados às ações da Petrobras, por meio de informações coletadas online baseadas em dados não estruturados (notícias/mensagens).

1.2 Objetivo Específicos

- Coletar os dados de texto (não estruturados) para a análise via *Twitter*.
- Realizar o pré-processamento do conjunto de dados obtido com a análise de objetos duplicados, limpeza e transformação;
- Comparar as características quantitativas e qualitativas dos grupos formados;
- Contribuir com a área de estudo no mercado financeiro brasileiro.

1.3 Justificativa

É necessário acompanhar o campo de estudo para qual uma área converge, uma vez que isso pode apontar que esta obtém melhores resultados. Com a afirmação de que “A literatura recente se afastou dessas abordagens tradicionais de medir os sentimentos do mercado com base em indicadores macroeconômicos ou financeiros.” (PARAMANIK; SINGHAL, 2020), e anteriormente Yadav *et al*. (2020), apontando que as finanças comportamentais modernas consideram tanto o sentimento como também o racional do investidor, diferente da teoria financeira clássica, faz se necessário estudos voltados para a análise de dados não estruturados com enfoque na alteração

dos sentimentos dos investidores de acordo com as notícias sobre o mercado financeiro.

Ademais, Yadav *et al*. (2020) também citam que não é mais como algumas décadas atrás que a principal questão era se o sentimento do investidor afeta os preços das ações, mas sim como quantificar esse sentimento e medir seu efeito. Sendo assim, ao levar esses pontos em consideração, fica evidente que realizar estudos na área da mineração de dados não estruturados pode contribuir tanto para os investidores como também para a formação de uma literatura mais sólida, comprovando a eficiência da aplicação desse método.

Katayama e Tsuda (2020) apontam que ao aplicarem o *machine learning* e analisar o sentimento dos investidores com base em notícias como estratégia de investimento, no curto prazo, foi obtido um bom desempenho. Além disso, Katayama e Tsuda (2020) também afirmam que investidores podem ter um desempenho melhor e superar o mercado ao utilizar esse método para processar grandes quantidades de dados.

1.4 Estrutura do trabalho

Após a exposição do contexto, objetivos, e justificativa do trabalho, expostos no capítulo introdutório, é apresentado o referencial teórico no Capítulo 2, onde a fundamentação abrange o funcionamento do mercado acionário brasileiro, possibilidade de manipular grandes quantidades de dados não estruturados na mineração de texto, meios utilizados para a normalização de textos como *stemming* e remoção de *stop words*, o que pode impactar na análise de sentimentos, funcionamento do *Bag-of-Words* e do *Term Frequency-Inverse Document Frequency*, para o tópico tarefa de agrupamento é abordado o algoritmo *K-means* e meios para determinar o número inicial de clusters, por fim, a técnica do PCA para redução de dimensionalidade é o último assunto abordado no referencial teórico.

O Capítulo 3 inicia pontuando sobre a fonte e meio de obtenção do conjunto de dados, mostrando sequencialmente os recursos utilizados, pontuando principalmente o Google Colaboraoty (*notebook*). As bibliotecas, como último tema, descreve as etapas realizadas na pesquisa.

Nos resultados e discussões (Capítulo 4), são trazidos o detalhamento da pesquisa que aborda a coleta de dados, pré-processamento, obtenção da matriz de

pesos, agrupamento antes e após o redimensionamento da matriz de pesos com aplicação do PCA. Outro tema abordado nesse tópico é a comparação dos resultados entre o os dois agrupamentos realizados por meio dos gráficos de dispersão, nuvens de palavras e análise de dados.

Na etapa final, é posto as considerações sobre a pesquisa realizada, além de sugestões para trabalhos futuros (Capítulo 5).

2 REFERENCIAL TEÓRICO

Nesta seção, será apresentado o embasamento teórico do trabalho, iniciando com a seção secundária com um breve texto sobre o mercado nacional de ações, seguida por uma passagem sobre mineração de texto, pré-processamento de texto, análise de sentimentos, *Bag-of-Words*, *Term Frequency-Inverse Document Frequency*, tarefa de agrupamento, finalizando com a subseção sobre a técnica de redimensionamento *Principal Componente Analysis*.

2.1 Notícias e especulações e o impacto no mercado de ações

A B3 (Brasil, Bolsa, Balcão) é a Bolsa de valores oficial do Brasil sendo para Canto (2020) um lugar centralizado onde se negociam parcelas do capital social de empresas de capital aberto, conhecidas como ações e nomeadas também de ativos ou papéis, além de abranger outros tipos de investimentos. Para ocorrerem as negociações das empresas listadas na bolsa é necessário que seja dia útil, durante o pregão eletrônico que ocorre das 10:00 às 17:00, onde as operações são realizadas por meio de uma corretora (CANTO, 2020).

A negociação desses ativos gera volatilidade no mercado, refletindo na variação de preço das ações e conforme diz Canto (2020) o preço desse ativo na Bolsa de Valores pode ser determinado por diferentes razões, entre elas as perspectivas de crescimento da empresa associada ao papel e especulação, a lei da oferta e demanda.

A notícia se faz responsável por boa parte dessas variações de preços dos ativos impactando diretamente no índice do Ibovespa. Exemplo disso, foi quando o *Chief Enterprise Officer* - CEO da empresa JBS S.A, Joesley Batista, realizou uma delação premiada para a Operação Lava Jato (Ministério Público Federal - MPF, 2014), um dia após a notícia ser divulgada, e antes do impacto da pandemia do COVID-19 na bolsa de valores: "O índice Ibovespa teve sua maior queda diária (-8,8%) desde a crise do *subprime* de 2008. Tal queda levou à ativação do *circuit breaker* para as ações na bolsa de valores brasileira, suspendendo as negociações financeiras e interrompendo os negócios por um período de 30 minutos" (MARIZ, 2020).

Pode-se citar também a notícia do rompimento da barragem de Mariana que teve grande impacto nas ações da empresa Vale, detentora de 50% das ações da Samarco (VALE, 2015) responsável pela barragem. Ademais, notícias de fusão de empresas também influenciam muito na volatilidade do preço de uma ação, como por exemplo, após a CIA Hering negar a proposta de fusão da Arezzo Indústria e Comércio SA, as ações da primeira valorizaram 28,13% enquanto o preço da segunda teve um aumento de 8,33% (INFOMONEY, 2021). Além de informações relevantes como estas mostradas nas mídias, podemos citar notícias sobre a divulgação sazonal dos demonstrativos financeiros de empresas de capital aberto - sociedade anônima onde os seus títulos, ações, são negociadas na bolsa de valores - que também impactam nos preços das ações na bolsa.

Sendo assim, os investidores enfrentam dificuldade para realizar operações que geram lucro. Além disso, "Com o advento do pregão eletrônico, muitas tecnologias foram aplicadas ao mercado de ações como os algoritmos de estratégias automatizadas.” (DUARTE *et al*., 2019).

2.2 Mineração de Texto

O processo de mineração de dados, conforme Charu (2013), trabalha com dois tipos de dados: dados orientados por dependência, que podem estar relacionados de maneira implícita ou explícita, e não orientados por dependência, sendo dados multidimensionais ou textos, que não possuem dependências específicas entre si.

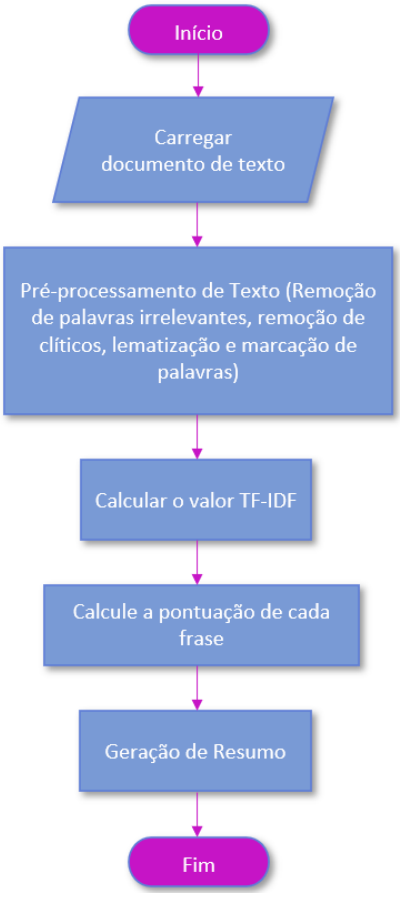
Esses dados de texto, de acordo com Charu (2015), têm um aumento gradual em seu volume com o decorrer do tempo, pois existe a facilidade em arquivar a fala e a expressão humana e, reforçando essa tendência, temos também o acréscimo contínuo na digitalização de bibliotecas e a onipresença da *web*.

Desta forma e considerando o fato desses dados serem não estruturados, Amo (2003) *apud* Morais (2007) aponta que a informação contida nestes textos, frases ou palavras não pode ser obtida de forma direta utilizando métodos tradicionais de consulta, sendo necessário aplicação de algoritmos computacionais que processam textos e identificam informações úteis e implícitas. Com isso, e diferentemente de dados numéricos (estruturados) que são convertidos de forma direta, quando necessário, os dados textuais (não estruturados) precisam de um pré-processamento de forma a quantificar as palavras (*strings*) contidas em um texto livre. Isto também

pode ser feito por meio de algoritmos de *deep learning,* para que seja contornado o problema de dimensionalidade na classificação de texto, onde palavras ou frases são mapeadas em representações contínuas em um espaço vetorial (LIU *et al*., 2020).

A Figura 1, representa um fluxograma simplificado do que ocorre com dados não estruturado com informações de textos, que após o início passa pela etapa de carregamento do documento, etapa de pré-processamento, onde apresenta a principal diferença entre trabalhar com dados numéricos e de texto, posteriormente é feito o cálculo do peso do *Term Frequency-Inverse Document Frequency* (TF-IDF), determinação dos pontos de cada frase, e por fim têm a geração de um resumo.

Figura 1 - Fluxograma de sumarização automática



Fonte: Christian (2016, p.288).

Ademais, de acordo com Fernandes (2018) a mineração de textos (*Text Mining*) é um Processo de Descoberta de Conhecimento em Texto, porém utilizando dados de entrada não estruturados. Morais (2007) aponta que com a finalidade de resolver

um problema ou uma situação, o descobrimento de conhecimento identifica informações relevantes agregando-as ao conhecimento prévio do usuário.

2.3 Pré-processamento de texto

Textos contêm muitos ruídos e palavras que não trazem informações relevantes, e mantê-las pode aumentar a dimensionalidade do problema, dificultando na sua resolução (HADDI *et al*., 2013).

Ademais, Munková *et al*. (2013) discorre que o pré-processamento de dados é a fase que mais demanda tempo para o processo de descoberta de conhecimento, depende muito da complexidade das fontes de dados utilizados, e influência na qualidade do resultado final.

Desta forma, ao trabalhar com dados de textos, e com o intuito de reduzir a dimensionalidade dos dados, reduzir os ruídos que não trazem informação, e consequentemente o esforço computacional, algumas técnicas e métodos são aplicados.

2.3.1 *Stemming*

O método de *stemming*, conforme apontam Soares *et al*. (2008), ajuda a reduzir a quantidade de termos, descritos também como *tokens*, ao transformar cada termo no radical que foi responsável por sua origem. Neste contexto, o *stemming* é utilizado para realizar a normalização linguística de um documento de texto, por meio do corte de sufixos ou prefixos de palavras como verbos e adjetivos para que esta seja reduzida à forma conhecida como *stem* (MARTINS *et al*., 2003).

Além disso, De Ávila e Soares (2013) interpretam que o *stemming* reduz os termos ao seu radical ou raiz, com a redução de variantes morfológicas como por exemplo formas singulares, plural e conjugações verbais. Como um exemplo disso, as palavras "conectar", “conectados”, “conectaram” e “conectei” podem ser transformadas para o mesmo *stem* “conect”.

É importante ressaltar que o algoritmo de *stemming* mais conhecido para remoção de sufixos em inglês é o de Porter (1980), que serviu como base para outros algoritmos serem adaptados e trabalharem com o idioma português, como exemplo o

RSLP *Stemmer* que é baseado em regras devido à grande variação existente na língua portuguesa (SOARES *et al*., 2008).

Desta forma, Martins *et al*. (2008) interpreta que esse algoritmo tem uma relação forte com o idioma no qual os documentos estão escritos. Porter (1980) pontua que a eficiência do algoritmo se degrada conforme ele se torna mais específico ao tentar minimizar a quantidade de *stems* diferentes para palavras com um mesmo radical. Kraaij *et al*. (1994) afirmam que existe a possibilidade de criar palavras inexistentes no idioma como efeito colateral.

2.3.2 Lematização

Similar ao método de *stemming*, pelo fato de “cortar” as palavras, este método se difere ao garantir que a forma comum de uma palavra exista no idioma, esta forma reduzida recebe o nome de lema, que é a forma canônica de um termo considerando o contexto em que ele foi utilizado (RIBEIRO, 2020).

Diogo e Ferreira (2019) dizem que a lematização realiza a simplificação das palavras em um documento de texto até sua forma mais elementar, igual aparece no dicionário. Exemplo disso, são as palavras “investi”, "investiremos" e “investiram” que podem ser reduzidas a “investir”.

2.3.3 *Stopwords*

O processo de eliminar *stopwords* trabalha com a remoção de palavras que ocorrem muitas vezes em um documento perdendo sua relevância, ou a remoção de termos que não contém informações importantes para o documento (MUNKOVÁ *et al*., 2013).

Existem diversas palavras que podem ser utilizadas como *stopwords*, cabendo ao usuário criar uma lista ou utilizar listas pré-existentes. A biblioteca *Natural Language Toolkit* (NLTK), definida por Loper e Bird (2002) como uma biblioteca de código fonte aberta utilizada comumente no aprendizado da linguagem natural, possui uma lista predefinida de stopwords. Como um exemplo de stopwords temos as palavras “era”, “depois”, “essas”, “haja”, “não” e “o”, porém as listas podem ser modificadas de acordo com o contexto analisado.

Haddi *et al*. (2013) exemplificam que ao trabalhar com o tema “filmes”, palavras como “filme”, “ator”, “atriz” e “cena” podem ser consideradas como stopwords, uma vez que essas não trazem muito significado para dados de resenhas de filmes. Sendo assim, fica evidente que as listas de stopwords podem variar de acordo com o tema abordado. Ressalta-se que o idioma do conjunto de dados interfere na lista de stopwords, posto que, existem variações diferentes nas línguas. Exemplo disso é a negativa do português e inglês, com o segundo apresentando mais variações como “is not”, “isn’t”, “cannot”, “can’t”.

No que diz respeito à contribuição para o algoritmo como um todo, Haddi *et al*. (2013) explicam que ao utilizar os métodos de stemming e remoção de stopwords no pré-processamento de um texto, a precisão da análise de sentimentos é melhorada.

2.4 Análise de sentimentos no mercado finaceiro

A palavra “sentimento” dentre outros significados, é a “Disposição afetiva em relação as coisas de ordem moral ou intelectual“ (FERREIRA, 2001, p. 631). Para Kinyua *et al*. (2021) sentimento é uma atitude subjetiva ou uma visão em relação a um evento ou situação, podendo ser referido também como uma opinião. Desta forma, no mercado financeiro, conforme também aponta Kinyua *et al*. (2021) a crença do risco de investimento e os supostos fluxos de caixa futuros de um ativo impactam o sentimento do investidor. De acordo com Antonakaki (2021), esse sentimento normalmente é uma variável que recebe valores específicos como “feliz” e “zangado” ou até mesmo valores como “positivo”, “negativo” e “neutro”. Além disso, uma única palavra é permitida a receber múltiplas atribuições de sentimentos, uma vez que uma variável pode ter inúmeros valores.

Os valores das variáveis podem ser, por exemplo, informações de um ativo, notícias e postagem em mídias sociais. Com relação ao último, Ali *et al*. (2017) descreve que para eventos de emergência, usuários compartilham e criam dados rapidamente por meio das redes sociais, funcionando como sensores humanos ao trazer informações subjetivas (sentimentos e opiniões) e uma relação de espaço-tempo. Ademais, Statista (2017) *apud* Alamoodi *et al*. (2021) afirma que isso ocorre devido ao tempo que os usuários ficam “online” nesses aplicativos.

Um ponto importante é que esses dados ficam em *sites*, na maioria das vezes gratuitos, como por exemplo o *Facebook*, *Twitter* e *blogs*, tornando interessante a coleta de dados por esses meios para uma posterior análise de sentimentos.

2.5 *Bag-of-Words*

A conversão de documentos de textos em instâncias com um número fixo de atributos, segundo Bramer (2007) pode ocorrer de inúmeras formas como por exemplo ao contar combinações de caracteres, palavras consecutivas ou ao numerar a aparição de frases específicas. O método *Bag-of-words* (BoW) é uma dessas formas que realiza o procedimento citado, mapeando um documento em um espaço vetorial e quantificando o número de aparições de uma palavra (KONONOVA *et al.*, 2021). Com isso, conforme explica Merlini e Rossini (2021), em um documento de texto a aparição de um termo faz seu valor no vetor diferente de zero, mais tarde esse valor pode ser utilizado como um peso, mostrando a importância do termo no documento.

As frases “O carro preto é bonito. O carro também é grande” e “O carro branco é grande e confortável” são exemplos mostrados na Tabela 1, após a aplicação do método BoW, onde são vetorizadas, com as *stopwords* removidas, para que o computador possa “interpretar”, e se necessário levar em consideração o peso de cada termo de acordo com seu número de repetições.

Tabela 1 – Exemplo de vetores

bonito	branco	carro	confortável	grande	preto	também
1	0	2	0	1	1	1
0	1	1	1	1	0	0

Fonte: Autoria própria (2021)

Para Seong e Nam (2021), com o que diz respeito à mineração de texto para previsão do preço de ações, BoW é a abordagem mais utilizada no pré-processamento na etapa de extração de recursos. Embora seja o modelo mais utilizado, o BoW ignora o arranjo dos parágrafos, ordem e combinação das palavras, seus significados e a utilização de pontos (BRAMER, 2016). Além de não conseguir identificar a importância de uma palavra em um texto e seu contexto, cria um obstáculo

que pode ser resolvido ao realizar a aplicação de um fator de normalização, em cada contagem de palavras (KONONOVA *et al.*, 2021).

2.6 *Term Frequency - Inverse Document Frequency*

O *Term Frequency-Inverse Document Frequency* (TF-IDF) pode ser utilizado em conjunto com o BoW, realizando-se a aplicação de um fator de normalização. A combinação destes dois modelos ajuda na identificação de informações relevantes ou classificação de um documento, onde o TF-IDF realiza a associação de duas métricas: (i) frequência de aparição em um documento, e (ii) a fração de documentos que contém a palavra (KONONOVA *et al*., 2021).

Além disso, Chatterjee *et al*. (2020) explica que uma palavra que aparece com frequência de forma repetida em um documento particular possivelmente é importante, por outro lado, se uma palavra aparece em todos os documentos, é menos importante. Dessa forma, essa lógica é obtida pelo TF-IDF, que dá pontuação às palavras funcionando como um proxy (elemento intermediário) (FELDMAN; SANGER, 2007).

Ademais, Bramer (2016) aponta que em relação a outros, esse método tem obtido melhor desempenho, e para realização do seu cálculo, o valor de um peso w_{ij} é obtido através da multiplicação das duas métricas, frequência inversa do documento vezes a frequência do termo, dado pela Equação 1.

$$w_{ij} = tfidf_{i,j} = tf_{i,j}\ x\ log_2\left(\frac{N}{df_i}\right) \quad (1)$$

Onde w_{ij} é o peso do termo i no documento j , o termo tf_{ij} representa o número total de ocorrências de i em j, e N o número total de documentos. O termo df_i é o número total de documentos que contem i, ou seja, a frequência de i (TRSTENJAK; MIKAC; DONKO, 2014).

Segundo Christian *et al*. (2016), esse peso classifica as palavras em ordem decrescente, recebendo um valor que varia de 0 a 1, para posteriormente calcular o valor de importância de uma frase, que é a soma dos pesos recebidos pelos substantivos e verbos que compõe a frase.

Desta forma, dependendo do programa escolhido pelo usuário, são selecionadas de 3 a 5 sentenças com maior valor de TF-IDF (CHRISTIAN; AGUS; SUHARTONO, 2016).

2.7 Tarefa de Agrupamento

Existe um grande número de algoritmos de agrupamento disponíveis na literatura. O *K-means*, é um desses e apresenta como vantagem o baixo esforço computacional por calcular apenas a distância dos pontos até os centroides. (ULFENBORG *et al*., 2021). Neste contexto, foi utilizado este algoritmo para agrupamento dos *tweets* avaliados neste trabalho.

2.7.1 *K-Means*

O algoritmo *K-means*, método não supervisionado para a separação de dados com base em suas características, tornou-se popular por convergir e estabilizar rapidamente a atribuição dos dados aos *clusters*. Isso acontece devido ao procedimento de otimização iterativa que associa aleatoriamente as informações a um dos *k-clusters* determinados previamente, buscando minimizar a soma quadrada das distâncias de cada ponto de dados ao seu centro de *cluster* atribuído (também conhecido por centróide), e se necessário, refaz o processo e redistribui os conjuntos de dados aos *clusters* mais próximos (CHEN *et al*., 2021).

Matematicamente, conforme Jiang *et al*. (2021), o algoritmo divide os dados de entrada x em k grupos, resultando no grupo $C = \{C1, C2, Ck\}$ satisfazendo da Equação 2:

$$\bigcup_{\tau=1}^{k} C_\tau = X, \qquad C_\tau \cap C_v = \Phi \tag{2}$$

Onde $1 \leq \tau \neq k$ e Φ é um conjunto vazio.

Jiang *et al*. (2021) ainda complementa que o número de *clusters*, denotado por k, precisa ser do conjunto de números naturais e definido pelo usuário, em que cada k inicialmente recebe posições aleatórias, para que os dados sejam atribuídos a um

cluster mais próximo. Deste modo, busca minimizar a soma do erro quadrático das distâncias (comumente a distância euclidiana), dada pela Equação 3:

$$\arg min\ C \left\{ \sum_{j=1}^{k} \sum_{X \in C_j} \left| x - \mu_j \right|^2 \right\} \tag{3}$$

Por último, Shrifan *et al*. (2021) aponta para atualização da posição dos centroides em cada iteração, então a média dos pontos de dados μ_i, dentro de cada *cluster* é calculada, em que N representa o número de pontos de dados dentro do enésimo *cluster*, sendo o cálculo definido pela Equação 4:

$$\mu_j = \frac{1}{N} \sum_{X \in C_j} X \tag{4}$$

Desta forma, é determinado o valor médio dos objetos do *cluster*, atualizado os centros do *cluster*, sendo continuado até não ocorrer mais alterações nos centróides ou ao atingir o número máximo de iterações, e por fim, o algoritmo retorna o melhor centro de cada *k-cluster* e os *clusters* C_j formados para $j = 1,2$ até k. (JIANG *et al*, 2021)

Como o *K-means* é baseado em aglomerados esféricos, em que os pontos de dados convergem em torno do centroide do cluster, ao final, os pontos dentro de um mesmo cluster tem suas semelhanças entre si aumentadas. (SHRIFAN *et al.*, 2021)

2.7.2 Número inicial de *clusters*

Ao utilizar o algoritmo *K-means* para separar clusters e seus respectivos componentes, embora seja um método de rápida convergência, enfrenta-se dois problemas iniciais, um desses problemas é a definição dos centroides iniciais de cada *cluster*, que ocorre de maneira aleatória (ABUALIGAH *et al*., 2020).

Como segundo problema, Fahim (2021) aponta para a determinação do número que deve ser atribuído a variável k, ou seja, o número de clusters total no qual os dados serão alocados, que é um parâmetro de partida do algoritmo *K-means*.

Apesar de existirem muitas ideias para resolver separadamente esses problemas, não existe uma solução única que resolva ambos os problemas simultaneamente de acordo com Fahim (2021). Desta forma, e considerando que para a linguagem Python o próprio *k-means* roda um algoritmo de otimização para inicialização dos centroides, o *Kmeans* ++, é buscado mitigar apenas os desvios causados pelo segundo problema, definir o número de *clusters*, por meio da utilização dos métodos de *Within Cluster Sum of Squares* (WCSS) e *Silhoutte Score* para auxiliar na tomada de decisão.

a) *Within Cluster Sum of Squares* (WCSS): traduzido como Soma dos Quadrados Intra-*cluster* e também conhecido por *Cluster Inertia*, é uma etapa do algoritmo *K-means*, que busca minimizar a soma dos erros quadráticos dentro do *cluster* (OLSON, 2015).

   Desta forma, o WCSS por meio do cálculo da distância Euclidiana, permite que seja gerado um gráfico que confronta os valores das somas dos erros quadráticos dentro dos *clusters* com o número inicial de clusters predefinido para iniciar o *K-means*, possibilitando o acompanhamento do desempenho de forma a mostrar o melhor número de *clusters*.

b) Silhueta: O método da Silhueta para Naghizadeh e Metaxas (2020) é um apoio para interpretar e validar os dados nos *clusters*. Por meio de um gráfico, este método representa o quão bem cada objeto se encontra em seu cluster, o valor da Silhueta varia de -1 até +1 e é definido pela Equação 5:

$$s(i) = \frac{a(i)b(i)}{\max\{a(i), b(i)\}} \tag{5}$$

   No qual $b(i)$ representa a dissimilaridade média mínima do enésimo ponto i para o *cluster* do qual não seja um membro, $a(i)$ indica a dissimilaridade média do enésimo ponto i com relação a todos os outros dados que compõem o mesmo *cluster*. Caso $S(i)$ esteja mais próximo de 1, indica que o ponto está corretamente em seu cluster, por outro lado, $S(i)$ perto de -1, manifesta que o ponto deveria estar agrupado em outro *cluster* (SUBBALAKSHMI *et al*., 2015).

Desta forma, Subbalakshmi *et al*. (2015) diz que a média $S(i)$ indica o quão corretamente os dados foram agrupados, auxiliando para definir o número de *clusters* que um conjunto de dados deve ter.

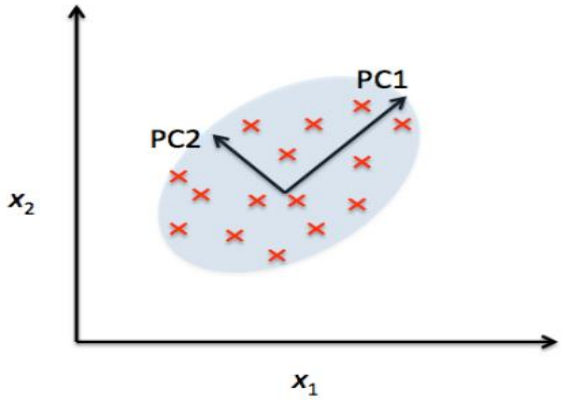
2.8 *Principal Component Analysis*

Ao realizar a tokenização de diferentes documentos de textos e alocá-los em um único conjunto de dados, criam-se vetores com grandes dimensões. Mohamed (2020) aponta que ruídos produzidos pela alta dimensionalidade podem afetar a precisão do algoritmo de agrupamento, entretanto, existem técnicas que são utilizadas para redução de dimensionalidade, e consequentemente de um possível ruído, como por exemplo a técnica *Principal Component Analysis* (PCA, traduzido como Análise dos Componentes Principais).

O PCA é uma técnica matemática que busca reduzir a dimensionalidade de um conjunto de dados que detém muitas variáveis correlacionadas, por meio da transformação em um novo conjunto de variáveis que não são inter-relacionadas e denominadas de componentes principais, ou *Principal Component* (PCs). Esses PCs são ordenados para que as primeiras variáveis novas retenham a maior parte da variação perdendo o mínimo de informação presente no conjunto de dados original (MISHRA *et al*., 2017).

Olson (2015) resume que o PCA procura direções com variações máximas e projeta-as em um novo subespaço de dimensão menor ou igual ao original, em que os PCs, são os eixos do novo subespaço e ortogonais entre si, são as direções de variação máxima. A Figura 2 mostra PC1 e PC2 como componentes principais e x_1 e x_2 são os eixos do recurso original.

Figura 2 - Componentes Principais



Fonte: Olson (2015, p. 128)

Aidoo *et al*. (2021) diz que os PCs são combinações lineares das variáveis originais com dimensão resultante menor, sendo obtidos a partir do cálculo da variância-covariância da matriz do conjunto de dados iniciais.

Mishra *et al*. (2017) pontua também que o PCA é uma ferramenta poderosa para analisar dados, podendo ser também utilizada para identificar padrões e destacar suas semelhanças e diferenças, principalmente quando se trata de um conjunto de dados com grande dimensionalidade.

Comparando técnicas de redução de dimensionamento, Mohamed (2020) pontua que uma das vantagens apresentadas pela técnica do PCA é sua característica de forçar os vetores dos componentes principais a serem ortogonais entre si. Além disso, dentre as técnicas avaliadas em sua pesquisa, mostrou melhores resultados se comparada à acuracidade e informação mútua normalizada.

3 METODOLOGIA

Esta seção descreve o fluxograma do trabalho realizado para se atingir ao objetivo proposto, identificando-se todos os passos de implementação dos algoritmos, obtenção e manipulação do conjunto de dados, bem como as ferramentas e linguagem de programação utilizada nesta pesquisa.

3.1 Descrição do conjunto de dados

Os dados do presente trabalho foram coletados da rede social *Twitter*, por meio da conexão com a plataforma *Twitter* API v1.1, que é voltada para programadores, sobre a empresa Petrobras e os termos mais comuns ligados diretamente a mesma.

3.1.1 *Twitter*

Durante as eleições presidenciais dos Estados Unidos da América no ano de 2008, foram registradas no *Twitter*, em média 5 mil *tweets* (publicações) por minuto. No ano seguinte, com a morte do astro do rock Michael Jackson, a rede social saiu do ar durante 30 minutos após o elevado número de buscas realizadas sobre “Michael Jackson” onde o *Twitter* identificou que o termo estava ligado a requisições automatizadas realizadas por vírus e *softwares* de espiões (GLOBO, 2009).

Posto isso, e por se tratar de uma das maiores redes sociais do mundo considerando o número de usuários ativos, confirma-se o impacto e a relevância que a rede social tem sobre a sociedade, principalmente no que diz respeito à disseminação de informações. Isso não é diferente quando o assunto é mercado financeiro, exemplo disso são as contas existentes de investidores, figuras públicas ou até mesmo gestoras, que comentam sobre o assunto, como a Marilia Fontes, Thiago Nigro, Luciana Seabra, entre outros.

Desta forma, sabendo da influência que o *Twitter* pode ter na sociedade e até mesmo no mundo das finanças, foram coletados *tweets* relacionados ao mercado financeiro, mais precisamente sobre a empresa de capital aberto Petróleo Brasileiro S.A. Um ponto importante, é que esses dados por advirem de uma rede social não são padronizados e muitas vezes estão na linguagem informal, carregados de

abreviações pelo fato de cada *tweet* poder conter apenas 280 caracteres. Além disso, é possível ser encontrados *emojis* e *emoticons* (mensagens em formatos de imagens ou ícones) junto a esses textos.

Mais precisamente, o *Twitter*, funciona como uma espécie de *microblogging* onde os usuários postam mensagens que podem conter *hashtags* para explicitar um tópico a ser discutido, fotos, vídeos e até mesmo *links* de outros *sites*, permitindo ainda que o usuário tenha interação com o outro por meio de uma resposta, compartilhamento, “curtida” (*like*), repostagem (*retweet*) e até mesmo uma menção a outro usuário (TWITTER INC., 2021).

3.1.2 *Twitter* API

Os *tweets* são visíveis e pesquisáveis por qualquer pessoa no mundo, mesmo que o usuário esteja apenas olhando as atualizações das mensagens. O *Twitter* coleta informações pessoais do usuário, como por exemplo o tipo de dispositivo que está utilizando e o IP (*Internet Protocol Address*, traduzido como endereço de protocolo da internet), ou ainda informações adicionais como localização (TWITTER INC., 2021).

Apesar disso e por se tratar de uma rede social pública, não é possível que qualquer pessoa consiga coletar dados livremente do *tweet*. Para realizar essa coleta é necessário realizar uma solicitação de abertura de conta, sujeito à aprovação de acordo com as informações passadas e objetivo de uso informado, na plataforma do Twitter API, voltada para desenvolvedores. Após receber a aprovação, a *Twitter* API cria *tokens* e chaves de acesso que devem estar na sua linguagem de programação para autenticação e acesso aos *tweets*.

A *Twitter* API possibilita que o programador tenha acesso a milhares de *tweets* passados e até mesmo em tempo real, acesso a perfis de usuários específicos, tendências geográficas, busca de palavras, entre outras ferramentas (TWITTER INC., 2021). Esses recursos podem ser acessados por meio da biblioteca *TwitterSearch* que possibilita trazer os dados de texto e numéricos para o *Python* no formato *Json* (*JavaScript Object Notation*).

É importante ressaltar que existem diferentes versões da *Twitter* API e que se diferem pela quantidade de *tweets* que podem ser extraídos em um período de tempo, ferramentas adicionais, e até mesmo versões pagas. Neste trabalho foi utilizada a versão 1.1 que é gratuita, oferecendo para fins acadêmicos um limite de 10 milhões

de *tweets* por mês em cada projeto ou 500 mil para projetos padrões, com taxas de solicitações de atualização variáveis (TWITTER INC., 2021).

3.1.3 Petrobras

A empresa de capital aberto Petróleo Brasileiro S.A. conhecida também por Petrobras, foi escolhida devido a algumas características apresentadas pela empresa, por ser mundialmente conhecida, ter alta liquidez atingindo uma média diária no volume negociado de aproximadamente 1,2 bilhões de reais no ano de 2020 e estando sempre as cinco primeiras empresas mais negociadas na B3, e isso diz respeito apenas a ação preferencial (INFOMONEY, 2021).

Desta forma, de maneira a filtrar assuntos ligados à companhia, foram utilizadas como palavras-chave de busca os termos “Petrobras”; sobre as ações os termos “PETR4” (ação preferencial) e “PETR3” (ação ordinária); e o termo “petróleo”.

3.2 Recursos utilizados

3.2.1 *Python*

O *Python*, linguagem de programação mais popular para a ciência de dados, foi escolhido por ser de fácil acesso, poderosa e de alto nível, oferecendo um ambiente prático onde é possível anotar e executar ideias facilmente, e ter ampla disponibilidade de bibliotecas complementares (OLSON, 2015). Como ambiente de trabalho e armazenamento de códigos de programação (*notebooks*), foi utilizado o *Google Colaboratory* (*Colab*) da *Google Research*, pois não necessita de nenhuma instalação prévia por parte do usuário, é executado no navegador, possibilita acesso a GPUs modernas, é de fácil compartilhamento e gratuito (COLAB, 2021).

3.2.2 Bibliotecas

O *Python* é uma linguagem de programação atrativa para cientistas de dados devido às bibliotecas disponíveis, que oferecem recursos como funções matemáticas, modelagem de matrizes multidimensionais, estruturação de dados, visualização de

dados, entre outros. Desta forma, seguem as bibliotecas auxiliares utilizadas na pesquisa:

a) *TwitterSearch*: disponibiliza recursos que permitem a conexão do *Python* com o *Twitter*, coletando dados e *tweets* conforme os parâmetros previamente definidos.
b) *Datetime*: possibilita que junto aos *tweets*, venha detalhadamente a hora em que este foi criado.
c) *Json*: Leitura de forma organizada de todos os parâmetros buscados no *Twitter*.
d) *Pandas*: utilizada para importação e exportação de dados em arquivos na extensão ".csv" e ".xls", possibilita também a transformação de conjunto de dados em matrizes chamadas de *dataframes*, sua manipulação interna e com outros *dataframes*, como concatenação, remoção de duplicatas, cálculos matemáticos, entre outros.
e) *Regular expression* (Re): trabalha com expressões regulares, ajudando na manipulação de dados textuais, realizando sua normalização (remoção de caracteres indesejados e transformação para minúsculo)
f) BS4: conhecida também como *BeautifulSoup*, ajuda na remoção de *links* no texto trabalhado.
g) *Spacy*: tem uma base própria de *Stopwords*, que auxilia o usuário na limpeza do conjunto de dados textuais.
h) *Unidecode*: trabalha também com a normalização de texto por meio da remoção de qualquer tipo de acentuação que esteja localizada sobre um caractere, independente se é gramaticalmente correto ou não.
i) *Natural Language Toolkit* (NLTK): muito utilizada quando o assunto é Processamento de Linguagem Natural (PLN), pois oferece um conjunto de bibliotecas capazes de realizar diversas tarefas. Exemplo disso, é o recurso que tem capacidade de transformar uma palavra em sua forma raiz ou até mesmo realizar a remoção de sufixos.
j) *Scikit-Learn*: é uma biblioteca de aprendizado de máquina, que possibilita realizar de forma simples a aplicação do BoW, TF-IDF e PCA, além de disponibilizar o algoritmo que realiza a clusterização, o *K-means* e seus auxiliares para tomada de decisão o WCSS e a pontuação da silhueta.

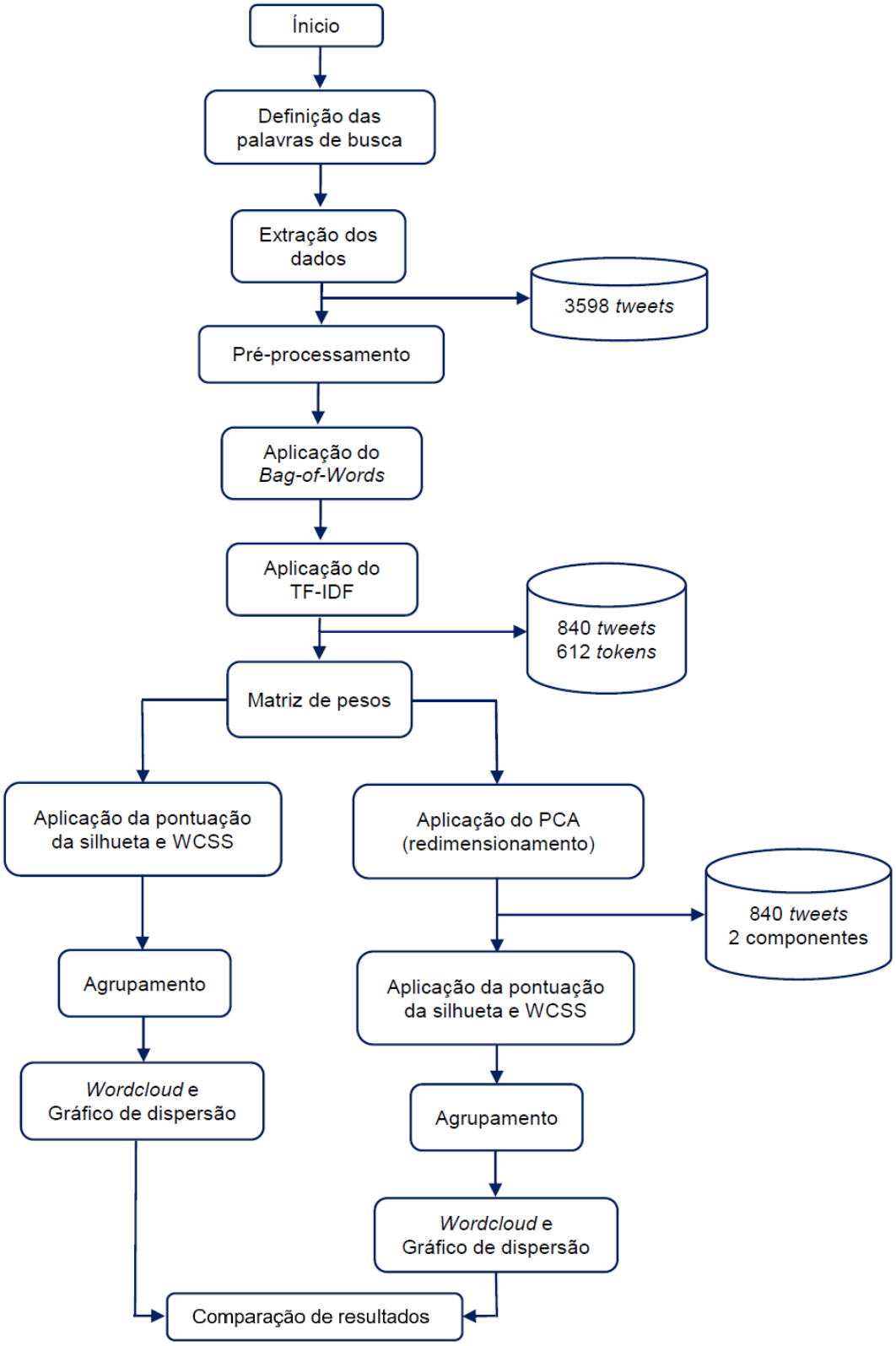
k) *Wordcloud*: auxilia na criação do gráfico de nuvens de palavras de acordo com a ocorrência dos termos nos documentos de textos.
l) *Matplotlib* e *Seaborn*: funcionam como bibliotecas auxiliares para trabalhar com estatística, visualização de dados e criações de gráficos.

Todas as informações mais específicas e os métodos incluídos em cada uma das bibliotecas podem ser encontrados em seus documentos originais, disponibilizados publicamente na internet.

3.3 Etapas da pesquisa

A Figura 3 representa o fluxograma resumido da pesquisa, seguida pela explicação detalhada dos procedimentos percorridos durante a pesquisa.

Figura 3 - Fluxograma das etapas da pesquisa



Fonte: Autoria própria (2021)

Inicialmente com auxílio dos recursos disponibilizados pela biblioteca *TwitterSearch*, foi realizada a busca de *tweets* por meio do parâmetro “palavras-chave”, ou seja, mensagens que continham palavras pré-definidas (neste caso, “Petrobras”, “petr4”, “petr3” e “petróleo”) ou interligadas às mesmas, por meio do *retweet* ou resposta ao *tweet* principal, que foram buscadas de acordo com o limite permitido para aquela versão da *Twitter* API. É importante ressaltar que além de colocar um parâmetro de busca, foi necessário definir as características buscadas. Neste caso foram as informações sobre data e hora de criação, identificação do usuário que publicou, texto do *tweet*, número de favoritos que o *tweet* recebeu, e por fim, a contagem de *retweets* recebidos.

Com o auxílio das bibliotecas *Json* e *Pandas*, esses dados foram transformados em um conjunto de dados com 3598 *posts*. A próxima etapa realizada foi o pré-processamento dos textos de cada *tweet*, iniciada pela remoção de *tweets* duplicados, transformação para minúsculo, remoção de *links*, caracteres indesejados e pontuações com recursos das bibliotecas Re e BS4. Após a base de dados ser pré-processada, ficaram apenas 895 *tweets* remanescentes devido a remoção de textos duplicados, permitindo a *Spacy* e *Unidecode* remover as *stopwords*, eliminar acentos, e transformar as palavras em seus respectivos radicais por meio da eliminação de sufixos.

Sequencialmente, os recursos da *Scikit-Learn* foram utilizados para aplicação do BoW e definição dos pesos de cada palavra por meio do TF-IDF. Como o TF-IDF resultou em uma matriz de pesos com 840 linhas (*tweets*) e 2355 colunas (todas as palavras do documento), foi realizada mais uma limpeza nos dados por meio da aplicação de duas funções: uma que eliminava palavras que apareceram menos do que três vezes no conjunto de dados inteiro, e a outra função eliminava números que não foram eliminados no pré-processamento, por estarem juntas as letras sem nenhum espaço. Isso resultou em uma matriz de pesos com 840 linhas e 612 colunas.

A realização do agrupamento ocorreu em dois conjuntos de dados distintos, sendo o primeiro apenas com as etapas anteriores realizadas e o segundo com as etapas anteriores e um redimensionamento por meio do PCA.

Antes da realização do primeiro agrupamento, foi realizada a aplicação do WCSS e a Pontuação da Silhueta na matriz de pesos, permitindo uma visualização gráfica que contribuiu para determinar o número de *clusters* (k), que nesse caso foi de cinco grupos. O agrupamento foi realizado com o algoritmo *K-means*, recurso do

Scikit-Learn, que recebeu como parâmetro de entrada a matriz de pesos (840 x 612) e k, resultando na separação de cada *tweet* em um cluster (identificados pelos números 0, 1, 2, 3 ou 4).

Com base nas etapas anteriores, foram criadas novas colunas no conjunto de dados, com a soma dos pesos de cada *tweet*, *cluster* atribuído ao *tweet*, peso médio e contagem de palavras analisadas de cada *tweet*. A biblioteca *Wordcloud*, foi utilizada para criar uma nuvem de palavras para cada um dos cinco *clusters*. E como última etapa, foi aplicado o PCA, redimensionando a matriz de pesos, reduzindo-a para apenas duas dimensões, possibilitando fazer um gráfico de dispersão para visualizar como os *tweets* estavam distribuídos de acordo com o *cluster* ao qual pertencia.

O segundo agrupamento, utilizando o PCA, redimensionou a cópia da matriz de pesos iniciais, transformando-a de 840 linhas x 612 colunas, para 840 linhas x 2 colunas. Novamente foram aplicados o WCSS e a Pontuação da Silhueta, porém dessa vez na nova matriz de pesos, que resultou na escolha de um k igual a 3. O *K-means* dessa vez recebeu como parâmetro de entrada a nova matriz de pesos (840 x 2) e a quantidade de três clusters, o que possibilitou agrupar os *tweets* nos *clusters* identificados por 0, 1 ou 2.

Posteriormente, foi realizada uma cópia do conjunto de dados inicial e adicionado a ela novas colunas: soma dos pesos de cada *tweet*, *cluster* atribuído ao *tweet*, peso médio e contagem de palavras analisadas de cada *tweet*. A nuvem de palavras foi criada para cada um dos três *clusters*. Uma vez que a matriz de pesos já estava redimensionada (840 x 2), foi criado um gráfico de dispersão com ela, identificando cada um dos três novos *clusters*.

Finalmente, foi possível comparar os resultados obtidos nos dois conjuntos de dados, ou seja, agrupamento realizado com e sem redimensionamento inicial da matriz de pesos obtida no TF-IDF.

4 RESULTADOS E DISCUSSÕES

Esta seção descreve de forma mais detalhada as etapas realizadas e os resultados obtidos em cada uma delas. Ademais, será realizada uma discussão mais profunda sobre os resultados obtidos com o primeiro e segundo agrupamento, ou seja, antes e depois do redimensionamento da matriz de pesos com uma *clusterização* para cada.

4.1 Detalhamento da pesquisa

4.1.1 Coleta de dados

Inicialmente para que houvesse conexão entre o *Colab* e a *Twitter* API, foi necessário criar um *app* (*application*) na plataforma de desenvolvedores obtendo então os *tokens* (*access token* e *acess token secret*) e as chaves de acesso (*consumer key* e *consumer secret*) que seriam informados para a *TwitterSearch* conseguir acesso aos dados pelo *Colab*.

Desta forma, foi possível buscar *tweets* por meio do recurso *TwitterSerachOrder*, informando as palavras-chave “Petrobras”, “Petr4”, "Petr3" e "Petróleo", e indicando o idioma desejado, que no caso foi o português do Brasil. O recurso *search_tweets_iterable* permitiu definir os atributos, ou seja, as características que viria junto a cada *tweet* localizado, que nesse caso foram a data e hora de criação, identificação do usuário que publicou, texto do *tweet*, número de favoritos que o *tweet* recebeu, e por fim, a contagem de *retweets* recebidos.

É importante ressaltar que os 3598 *tweets* foram coletados no dia 22 de setembro de 2021 com um intervalo de tempo que abrangeu das 23 horas do dia 21 de setembro até as 18 horas e 50 minutos do dia 22 de setembro de 2021, datas escolhidas devido à grande volatilidade do preço das ações no período e notícias como a queda do preço do barril de petróleo no dia 20, e dia 22 a notícia da renúncia do presidente da Transpetro, empresa da Petrobras voltada para o transporte e logística de petróleo e derivativos. Por último, *Json* e *Pandas* auxiliaram na transformação dos dados extraídos para um conjunto de dados organizado com 3598 posts e cinco colunas com as características filtradas no início.

4.1.2 Pré-processamento

O pré-processamento de dados textuais foi uma das etapas mais importantes do desenvolvimento, uma vez que ao trabalhar com mensagens de rede social, textos informais não são padronizados, como o termo “Petróleo”, que pode variar como “PETRÓLEO”, “Petróleo”, “petróleo” e “petroleo”, uma vez que o algoritmo diferencia letras maiúsculas, minúsculas e acentos. A solução para isto, foi normalizar removendo todas as variações chegando nesse caso ao termo “petroleo”, diminuindo o tamanho do banco de dados, e consequentemente o esforço computacional. Além disso, a limpeza de palavras que não causam impacto no contexto ou não trazem informações é outro procedimento importante. Os seguintes passos a passos foram seguidos:

a) Remoção de *posts* duplicados com o recurso *drop_duplicates*;
b) Transformação das mensagens em minúsculo, remoção de caracteres indesejados, incluindo *emojis*, *emoticons* e pontuações, com auxílio da Re;
c) Remoção de *links* com a BS4 e Re;
d) Remoção de *stopwords* com base na biblioteca *Spacy* que contempla um total de 413 palavras, com alguns exemplos listados na Tabela 2;
e) Remoção de acentos com o auxílio da biblioteca *Unidecode*;
f) Remoção de sufixos com o recurso RSLP *Stemmer* da biblioteca NLTK.

Tabela 2 - Exemplos de *Stopwords* da biblioteca *Spacy*

Coluna 1	Coluna 2	Coluna 3	Coluna 4
agora	algumas	daqueles	de
do	muito	não	nas
nenhum	outro	portanto	pouco

Fonte: Autoria própria (2021)

Desta forma, após a normalização dos dados, o conjunto ficou com 840 *tweets* e seis colunas, redução significativo na quantidade de linhas devido os *posts* duplicados, e uma coluna a mais que representa os novos *tweets* após sua limpeza, exemplificado na Tabela 3.

Tabela 3 - Antes e depois da normalização de *tweets*

Data	*Tweet* original	Número de Favoritos	Número de *retweets*	*Tweet* após normalização
2021-09-22 14:40:22	Hoje, completamos 40 dias sem reajuste de preços da gasolina A nas refinarias da Petrobras e mesmo assim o preço aumentou… https://t.co/0AyxC9vqyy	70	10	hoj complet 40 dia reajust prec gasolin refin petrobr prec aument
2021-09-22 10:23:00	Lembra como o volume estava forte? PETR4 agora tá projetando só R$69,6 MM de ações ... 😕	25	5	lembr volum fort petr4 projet 69,6 acoe
2021-09-22 10:00:02	Os preços da gasolina praticados pela Petrobras hoje têm defasagem média de 6% em relação aos preços internacionais… https://t.co/ygcYh5g5X2	53	64	prec gasolin pratic petrobr hoj defasag medi 6 prec internacion

Fonte: Autoria própria (2021)

4.1.3 Obtenção da matriz de pesos

Nessa etapa, foi trabalhado apenas com os *tweets* pré-processados, deixando de lado o seu formato original e seus atributos. A biblioteca *Scikit-Learn* foi essencial para o BoW e TF-IDF conforme explicados sequencialmente:

a) BoW: utilizando os recursos *CountVectorizer* e *fit_transform*, foi possível realizar a *tokenização*, permitindo criar o BoW, ou seja, um novo conjunto de dados com os termos separados. Essa matriz resultante, tinha 840 linhas representando os *tweets*, 2355 colunas representando todos os termos, e os valores sendo o número de repetições n que um termo i apareceu em um *tweet* j. Ademais, foi possível realizar também a contagem de vezes que os termos repetiram no conjunto de dados. A Tabela 4 representa os quatro primeiros termos que mais reincidiram nos *tweets* pré-processados.

b) TF-IDF: desta vez, utilizando o recurso *TfidfVectorizer* e novamente o *fit_transform*, foi criada uma nova matriz similar à gerada pelo BoW, porém dessa vez com os valores sendo os pesos atribuídos pelo algoritmo do TF-IDF, que levou em consideração a ocorrência de um termo em um

documento e a quantidade de documentos que o contém, dado pela Equação 1.

c) Segunda limpeza de dados: Com aplicação do BoW e o TF-IDF, foi possível visualizar todos os termos em colunas, suas repetições e pesos, deixando evidente muitos números que apareceram uma vez em um único *tweet*, e inúmeros termos que apareceram poucas vezes no conjunto de dados resultando em pesos insignificantes. Como esses casos não iriam impactar no agrupamento, duas funções realizaram suas remoções. A primeira removeu todos os números considerados como *tokens*, ou seja, separados em colunas, enquanto a segunda função removeu os termos que ocorreram menos do que três vezes no conjunto todo, número definido de forma empírica. Desta forma, foi obtido um novo conjunto de dados que tinha 840 linhas e 612 colunas, uma redução de aproximadamente 74% no total de termos. É importante ressaltar que essas funções foram aplicadas apenas na matriz de pesos gerada pelo TF-IDF, que seria utilizada para as próximas etapas da pesquisa.

Tabela 4 – Termos com maior repetição nos *tweets* pré-processados

Token	**Número de repetições**
petrobr	286
petrole	234
brasil	78
prec	64

Fonte: Autoria própria (2021)

Sequencialmente, com auxílio da matriz de pesos, foi aplicada o recurso soma e média, para obter os pesos de cada *tweet,* contagem de palavras e média dos pesos, sendo então, adicionados ao conjunto de dados iniciais, resultando em uma matriz com o mesmo número de linhas, porém com nove colunas (data de criação, identificação do usuário, *tweet*, número de favoritos, contagem de *retweets*, tweets pré-processados, soma dos pesos, média dos pesos e contagem de palavras).

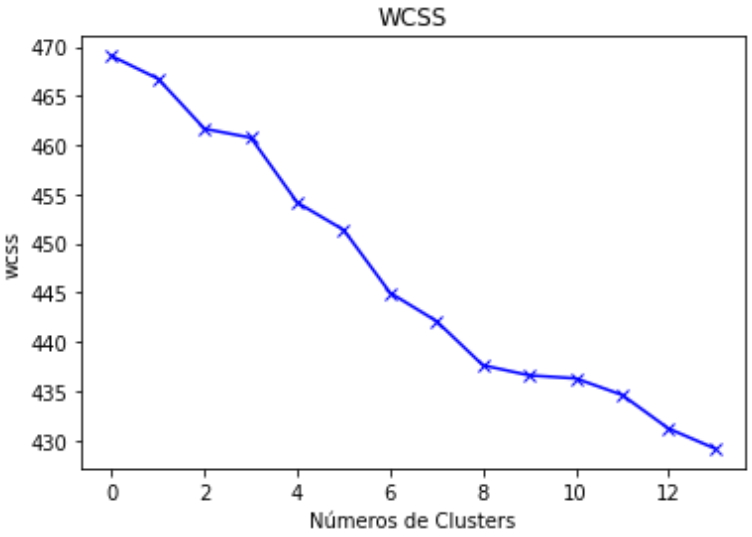
4.1.4 Quantidade de *clusters* para o primeiro agrupamento

Antes da realização do agrupamento foram aplicados os métodos da WCSS e pontuação da silhueta para ajudar na definição do número de *clusters*, tendo como entrada a matriz de pesos reduzida (840 x 612).

a) WCSS: com o recurso *Kmeans* foi possível obter as distâncias dos elementos até o seu centróide, variando a quantidade de grupos, que nesse caso foi pré-definido como sendo de k igual a 2 até igual a 15. Neste método, busca-se minimizar o valor do WCSS, e sempre o menor valor vai ser obtido quando o número de *clusters* for igual ao número de variáveis que precisam ser agrupadas, ou seja, para 840 *tweets*, o melhor cenário seriam 840 *clusters*. A Figura 4 apresenta o desempenho do WCSS.

b) Pontuação da Silhueta: o recurso *score_silhueta* e novamente *Kmeans*, permitiu verificar por outro método se os *tweets* estão bem agrupados em seus *clusters*. Nesse caso, utiliza-se o maior escore, a Figura 5 mostra o seu desempenho.

Figura 4 - Resultado do WCSS



Fonte: Autoria própria (2021)

Figura 5 - Desempenho Pontuação da Silhueta

Fonte: Autoria própria (2021)

Sendo assim, foi possível verificar que um k igual a 5, apresentou bons valores tanto no método do WCSS como também na pontuação da silhueta. Para esta pesquisa, não seria interessante abordar um k maior que cinco mesmo com melhores resultados, pois dificultaria as análises de cada grupo, e a princípio, k igual a 3 que apresentou uma boa pontuação, resultou em apenas em um grupo contendo aproximadamente 99% de todos os *tweets,* não sendo uma separação interessante para análises posteriores.

4.1.5 Primeiro agrupamento

Nesta etapa os parâmetros de entrada para o algoritmo *K-means* foram a matriz de pesos reduzida (840 x 612) e o número de *clusters*, que nesse caso foi cinco. A Tabela 5 mostra a quantidade de *tweets* que ficou em cada cluster (0, 1, 2, 3, 4).

Tabela 5 - Resultado do primeiro agrupamento

Nº do cluster	Quantidade de tweets	Representatividade com relação ao total (%)
0	12	1,43%
1	502	59,76%
2	37	4,40%
3	231	27,50%
4	58	6,90%

Fonte: Autoria própria (2021)

Com base nesses resultados, o conjunto de dados inicial passou a ter também o cluster de cada *tweet* ficando com o mesmo número de linhas (840) porém agora com 10 colunas, incluindo a data de criação, identificação do usuário, *tweet*, número de favoritos, contagem de *retweets*, *tweets* pré-processados, soma dos pesos, média dos pesos, contagem de palavras e *cluster*.

4.1.6 Aplicação do PCA

Antes da realização do segundo agrupamento, foi necessário aplicar o PCA para diferenciar os agrupamentos e possibilitar a comparação entre eles. Para isso, foi feita uma cópia da matriz de pesos obtida do TF-IDF após a aplicação das funções de remoção de palavras e números, com pouca aparição no conjunto de dados. Nesta cópia pré-processada (840 x 612), com auxílio do recurso PCA fornecido pela biblioteca *Scikit-Learn*, foi realizado o redimensionamento da matriz em suas colunas j, definindo inicialmente que número de PCs seria igual a dois, resultando então em um conjunto de dados do PCA com 840 linhas e 2 colunas.

4.1.7 Quantidade de *clusters* para o segundo agrupamento

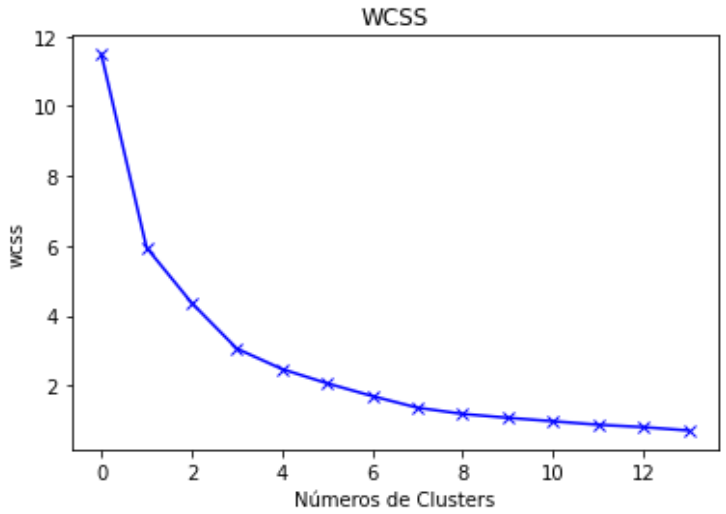
Novamente foi necessário definir o número de *clusters* antes da realização do agrupamento, uma vez que a nova matriz seria o conjunto de dados com PCA.

a) WCSS: foi mantida a variação inicial no número de *clusters* (2 até 15) diferenciando a entrada de dados, que para este caso foi os dados com

PCA. A Figura 6 mostra o desempenho obtido para a matriz de pesos redimensionada.

b) Pontuação da Silhueta: também recebeu como entrada a nova matriz de pesos redimensionada, buscando maximizar o escore conforme mostra a Figura 7.

Figura 6 - Resultado do WCSS nos dados com PCA



Fonte: Autoria própria (2021)

Figura 7- Desempenho Pontuação da Silhueta nos dados com PCA

Fonte: Autoria própria (2021)

Levando em consideração os resultados obtidos nos dois métodos, foi definido que o número de *clusters* seria igual a três. Embora $k = 2$ fosse um bom resultado

tanto no método do WCSS como também no método da silhueta, ao realizar o agrupamento 98% dos *tweets* ficaram em um grupo e apenas 2% no outro, dificultando a análise dos resultados.

4.1.8 Segundo agrupamento

Diferente do primeiro agrupamento, os parâmetros de entrada para o algoritmo *K-means* foram a matriz de dados com PCA (840 x 2) e o número de *clusters* igual a três. A Tabela 6 mostra a quantidade de *tweets* que ficou em cada *cluster* (0, 1 e 2).

Tabela 6 - Resultado do segundo agrupamento

Nº do cluster	Quantidade de tweets	Representatividade com relação ao total (%)
0	581	69,17%
1	240	28,57%
2	19	2,26%

Fonte: Autoria própria (2021)

Com base nesses resultados, foi criada uma cópia do conjunto de dados que tinha todas as informações do primeiro agrupamento, removendo-as e adicionando as novas informações de acordo com o segundo agrupamento.

4.2 Comparação dos resultados

Com o intuito de facilitar a comparação entre os agrupamentos e seguindo as próximas etapas da pesquisa, foram criados gráficos de dispersão e *Wordcloud* para os dois agrupamentos, e realizadas análises quantitativas e qualitativas.

4.2.1 Gráfico de dispersão

No caso do primeiro agrupamento, para criar um gráfico de dispersão com auxílio da biblioteca *Matplotlib* foi necessário aplicar o PCA em sua matriz de pesos, correlacionando os resultados com o *cluster* que havia sido atribuído a cada *twitter*, conforme é exemplificado pela Tabela 7. Por outro lado, para o segundo agrupamento

já havia a matriz de dados com PCA, foi somente necessário colocar os *clusters* atribuídos a cada *tweet*.

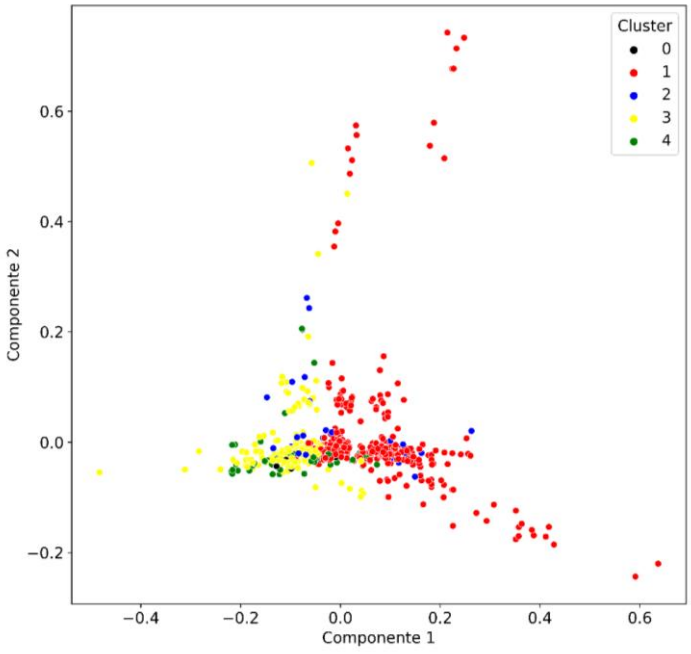
Tabela 7- Primeiros *tweets* e seus componentes

Tweet	Componente 1	Componente 2	Cluster
0	-0,162564	-0,018367	3
1	-0,007267	0,002660	3
2	0,186110	-0,042650	3
3	-0,038084	-0,019751	3

Fonte: Autoria própria (2021)

De maneira a facilitar a localização espacial dos *tweets* de acordo com o seu grupo, a Figura 8 possibilita visualizar o domínio dos pontos em vermelho (*cluster* 1), predominantemente localizados no primeiro quadrante, enquanto que os pontos em amarelo (*cluster* 3), que detém a segunda maior quantidade de *tweets*, estão localizados em maior número no lado negativo do eixo x. Em síntese, 502 *tweets* estão em posição oposta a 231 outros *tweets*, evidenciando que eles têm características diferentes entre si. Os grupos 0, 2 e 4, são mais difíceis de identificar pela pequena quantidade de dados que contemplam, e pelo fato de alguns pontos coincidirem com os pontos dos dois grupos com maior número.

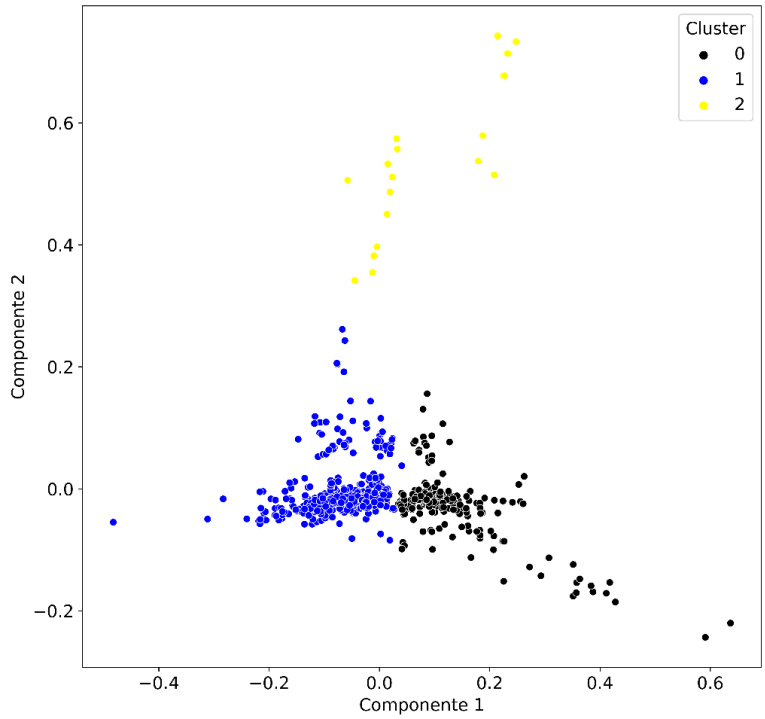
Figura 8 - Agrupamento 1



Fonte: Autoria própria (2021)

O agrupamento 2, mostrado na Figura 9, também aponta a divergência de quadrantes entre os dois grupos com maior quantidade de *tweets* representados pelos pontos pretos e azuis. Nesse caso, diferentemente do agrupamento um, o grupo com mais *tweets*, 581, aparenta ter menos *tweets* que o grupo com 240. Isso acontece porque no *cluster* 1 os dados estão mais espalhados, enquanto que no *cluster* 0, os dados estão mais juntos e se sobrepondo, ou seja, bem próximos aos seus centróides, explicitando mais características em comum dentro do mesmo *cluster*. Os pontos em amarelo, que representam uma pequena quantidade de *tweets*, mostra dados mais deslocados com relação aos outros e aponta para uma menor similaridade *intra-cluster*.

Figura 9 - Agrupamento 2



Fonte: Autoria própria (2021)

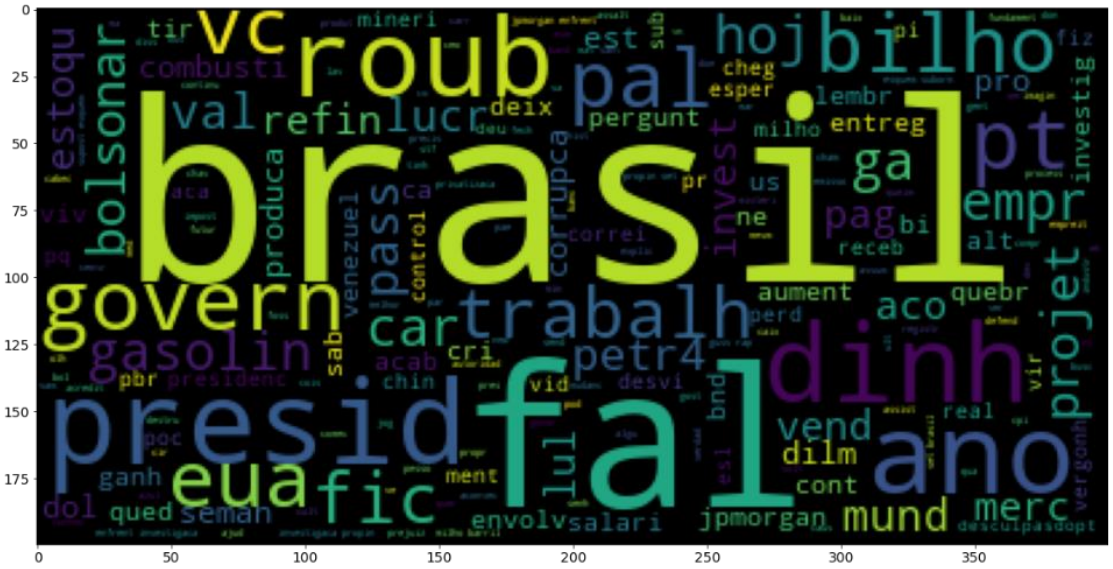
4.2.2 *Wordcloud*

Com o intuito de verificar quais palavras foram mais incidentes em cada um dos clusters e assim chegar a uma possível diferenciação de temas, foram geradas as *Wordclouds*, (em português, “nuvens de palavras”), para cada um dos *clusters*, tanto para o primeiro agrupamento, como também para o segundo, focando apenas nos grupos com maiores quantidades de *posts*. É importante lembrar que para este tipo

de visualização, ao definir o tamanho da palavra, o algoritmo considera o número de repetições que a palavra teve no conjunto de dados, ou seja, uma palavra por exemplo que aparece 100 vezes, terá o dobro do tamanho da palavra que repetiu 50 vezes.

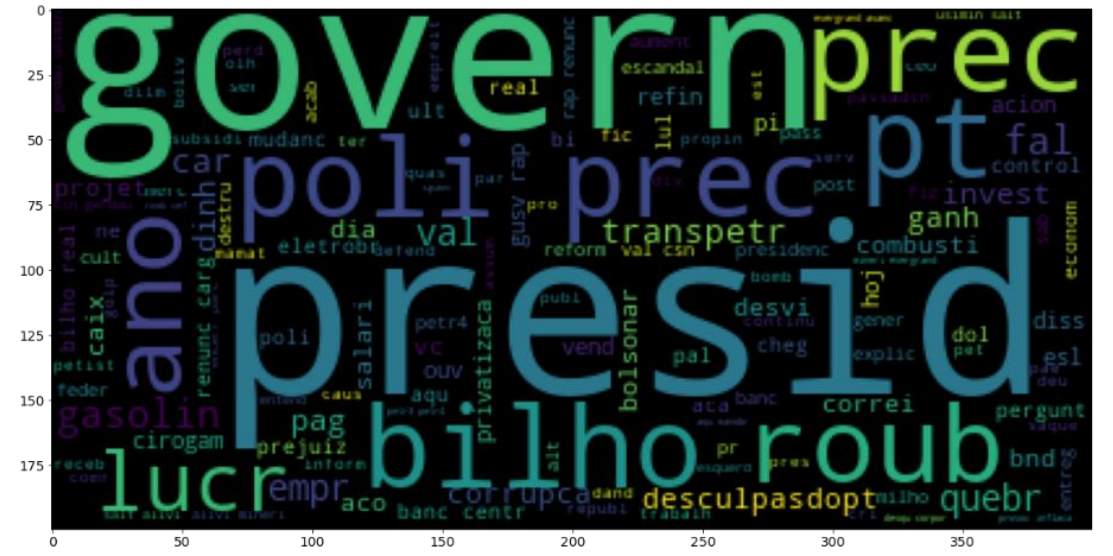
a) Primeiro agrupamento: No caso do grupo 1, como mostrado na Figura 10, as palavras “Brasil”, “governo”, “roubo”, “bilhões”, “presidente”, “Bolsonaro”, “dinheiro” e “falir” foram mais comuns. O grupo 3, Figura 11, teve grande incidência nas palavras “presidente”, “governo”, “bilhões”, “roubo”, “política”, “preço” e “lucro”. Isso mostra que mesmo os clusters 1 e 3 terem apresentados grandes diferenças em suas localizações espaciais, para a nuvem de palavras, ficou difícil associar cada cluster a um tema diferente, uma vez que as palavras mais incidentes foram similares ligadas diretamente a empresa, governo e ilegalidades. É importante ressaltar que em ambos os clusters a palavra mais incidente foi “petrobr”, ou seja, com maior tamanho na nuvem de palavras, porém como se tratava da palavra utilizada para fazer a busca foi removida para enfatizar outros termos.
b) Segundo agrupamento: O *cluster* 0, apresentado por nuvens de palavras da Figura 12, pode ser ligado ao assunto de comercialização e produção de petróleo, tendo como mais incidente as palavras “combustível”, “produção”, ”EUA”, “Brasil”, “estoque”, “gasolina” e “queda”. A Figura 13, representa as palavras “Brasil”, “governo”, “presidente”, “Bolsonaro”, “investigação”, “roubo”, “dinheiro” e “pt”, como sendo as palavras mais repetidas no *cluster* 1, relacionadas às ilegalidades e política. Neste caso, para ressaltar as outras palavras, foram removidas as palavras de busca “petróleo” mais incidente no *cluster* 0 e a palavra “Petrobras” mais incidente no *cluster* 1.

Figura 10 - Cluster 1 do primeiro agrupamento



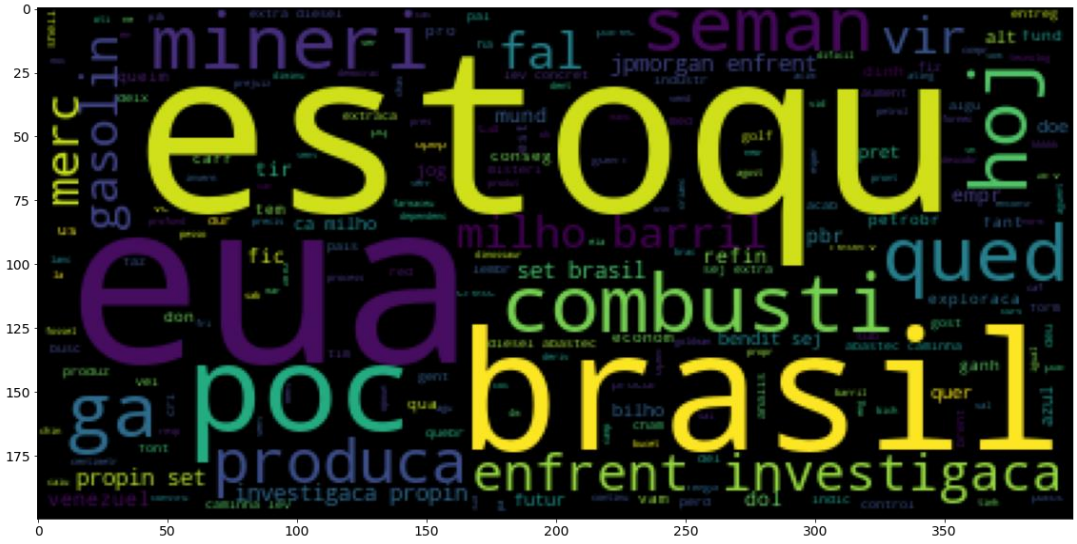
Fonte: Autoria própria (2021)

Figura 11 - Cluster 3 do primeiro agrupamento



Fonte: Autoria própria (2021)

Figura 12 - Cluster 0 do segundo agrupamento



Fonte: Autoria própria (2021)

Figura 13 - Cluster 1 do segundo agrupamento

Fonte: Autoria própria (2021)

4.2.3 Conjunto de dados

Após aplicação do TF-IDF e realização do agrupamento 1 foi possível agregar ao conjunto de dados novos valores, conforme citado anteriormente, como a contagem de palavras e seus pesos. O mesmo ocorreu para o agrupamento 2, que

recebeu redimensionamento em sua matriz de pesos. Isso possibilitou obter dados para uma outra análise quantitativa.

A Tabela 8, referente ao primeiro agrupamento, mostra que as variáveis (peso, *tokens*, nº de favoritos e *retweets*) não apresentam uma relação de proporcionalidade direta ou inversa com relação ao *cluster* e sua quantidade de *posts*.

Tabela 8 - Primeiro agrupamento

Cluster	***Tweets***	**Peso (w_{ij})**	***Tokens***	**N.° de favoritos**	***retweets***	**Média (Pesos)**	**Média (*tokens)***	**Média (favoritos)**	**Média (*retweets)***
0	12	23,3	73	5	1822	1,94	6,1	0,4	151,8
1	502	790,0	2653	1799	3118	1,57	5,3	3,6	6,2
2	37	72,3	245	113	171	1,95	6,6	3,0	4,6
3	231	406,9	1385	724	5305	1,76	6,0	3,1	22,9
4	58	118,3	402	89	607	2,04	6,9	1,5	10,5

Fonte: Autoria Própria (2021)

Por outro lado, a Tabela 9, relacionado ao segundo agrupamento, mesmo com diferenças pequenas, o *cluster* e sua quantidade de *tweets* tiveram relação inversa com os pesos e quantidade de palavras, ou seja, os *tweets* com menor média de pesos e de *tokens* foram agrupados no mesmo *cluster* e, conforme essas médias foram aumentando, os *tweets* foram agrupados em outro *cluster*.

Tabela 9 - Segundo agrupamento

Cluster	***Tweet***	**Peso (w_{ij})**	***Tokens***	**N.° de favoritos**	***retweets***	**Média (pesos)**	**Média (*tokens)***	**Média (favoritos)**	**Média (*retweets)***
0	240	398,97	1299	358	299	1,7	5,4	1,5	1,2
1	581	937,21	3298	2266	10695	1,6	5,7	3,9	18,4
2	19	49,0	161	106	29	2,6	8,5	5,6	1,5

Fonte: Autoria Própria (2021)

4.3 Desempenho

Em suma, é possível verificar que o segundo agrupamento, após aplicação do PCA na matriz de pesos obtidas pelo TF-IDF, teve melhor desempenho na separação de *clusters*, uma vez que para o caso do gráfico de dispersão os dados ficaram visualmente melhores, divididos e concentrados em seus centróides apontando para

uma melhor minimização das distâncias *intra-cluster*, ou seja, dados bem alocados em seu grupo. No que diz respeito a *Wordcloud*, ficou mais evidente as diferenças de assuntos contidas em cada *cluster* pelas palavras mais recorrentes em cada um deles. Por último, a análise quantitativa final mostrou que apenas para o segundo agrupamento é possível tirar breves conclusões, mesmo estas não tendo robustez o suficiente para validarem o agrupamento realizado.

5 CONCLUSÃO

O presente trabalho, utilizando o aprendizado não supervisionado, buscou agrupar *tweets* relacionados às ações da Petrobrás, e para tal, foi necessário explorar aplicações da mineração de texto em notícias do mercado financeiro, além de buscar maneiras para extração de *posts* e sua normalização. Posto isso, foi possível agrupar os dados e comparar tais grupos com gráficos de dispersão, nuvens de palavras e análise de dados.

Dentro desse contexto, inicialmente para a coletada de dados a *Twitter* API teve um papel importante para a pesquisa, sequencialmente com o estudo de trabalhos relacionados foi possível determinar como o pré-processamento de texto seria realizado e definir dois caminhos importantes para serem abordados, o primeiro com relação ao agrupamento realizado diretamente da matriz de pesos obtidas pelo TF-IDF, e o outro o agrupamento realizado a partir da matriz de pesos redimensionada com o PCA.

Desta forma, foi possível obter os resultados e concluir que aplicação do PCA na matriz de pesos antes do agrupamento trouxe um desempenho mais satisfatório do que realizar o agrupamento diretamente com os dados da matriz de pesos original, permitindo uma fácil visualização e diferenciação dos grupos no gráfico de dispersão, exaltando a similaridade dos dados de um mesmo *i*, ou seja, evidenciando a menor distância entre eles e consequentemente sobrepondo menos os dados de outro grupo. Além disso, no segundo agrupamento com o que diz respeito às nuvens de palavras, foi possível supor os temas mais recorrentes em cada um dos *clusters*.

Sendo assim, essas técnicas aplicadas podem ser úteis para um investidor, uma vez que em um intervalo de tempo muito pequeno é possível agrupar grandes quantidades de textos relacionados a uma empresa e supor os temas mais discutidos entre os textos, o que não seria possível de se realizar por uma única pessoa de maneira rápida. Além disso, caso o investidor for um especialista do tema, ele pode entender se as notícias envolvem sentimentos positivos ou negativos e quais seriam os impactos nos preços das ações, colaborando para uma tomada de decisão melhor.

Com o que diz respeito aos futuros trabalhos na área, explorar outros algoritimos de agrupamento para realizar comparação com *k-means*, pesquisar outros métodos de redimensionamento que estão disponíveis na literatura tais como o *Non Negative Matrix Factorization* e *Singular Value Decomposition*, são pontos que podem

melhorar significativamente os resultados e disponibilizar novos conhecimentos para a literatura.

Por fim, uma limitação apresentada no trabalho, é que ao realizar o pré-processamento dos dados, mesmo apresentado muito significado principalmente quando o assunto é rede social, os *emojis* e *emoticons* são removidos, o que pode ser tratado de maneira diferente, pois para a computação moderna existem caracteres que representam essas mensagens em formatos de imagens e ícones, possibilitando que sejam calculados os seus pesos pelo TF-IDF e consequentemente trazendo um resultado diferente para o trabalho. Além disso, por estar na lista de *stopwords*, a palavra “não” também é removida, abordagem que poderia ser diferente ao trabalhar com n-grama, trazendo outro resultado para o trabalho.

outro ponto importante, é que uma das limitações apresentada na pesquisa se trata da forma como é definido os temas abordados em cada *cluster*, partindo do pressuposto que as palavras mais incidentes definem o principal assunto de um grupo. Desta forma, contatar alguns especialistas sobre a temática estudada para validar os agrupamentos realizados, pode ser um diferencial ao confirmar ou invalidar o caminho seguido. Outra

REFERÊNCIAS

ABUALIGAH, L.; DIABAT, A.; GEEM, Z. W. A comprehensive survey of the harmony search algorithm in clustering applications. **Applied Sciences (Switzerland)**, v. 10, n. 11, p. 1–26, Mai. 2020.

AIDOO, E. N. et al. Geographically weighted principal component analysis for characterising the spatial heterogeneity and connectivity of soil heavy metals in Kumasi, Ghana. **Heliyon**, v. 7, n. 9, p. e08039, Set. 2021.

ALAMOODI, A. H. et al. Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review. **Expert Systems with Applications**, v. 167, p. e114155, 2021.

ALI, K. et al. Sentiment Analysis as a Service: A Social Media Based Sentiment Analysis Framework. **Proceedings - 2017 IEEE 24th International Conference on Web Services, ICWS 2017**, p. 660–667, Jun. 2017.

ALPHABET INC. **Google Colaboratory**. Disponível em: https://colab.research.google.com/. Acesso em: 19 out. 2021.

ANTONAKAKI, D.; FRAGOPOULOU, P.; IOANNIDIS, S. A survey of Twitter research: Data model, graph structure, sentiment analysis and attacks. **Expert Systems with Applications**, v. 164, p. 114006, Set. 2021.

BRAMER, M. **Principles of data mining**. 3. ed. New York: Springer, 2016.

B3. Bolsa, Brasil e Balcão. **B3 divulga estudo sobre os 2 milhões de investidores que entraram na bolsa entre 2019 e 2020**. Disponível em: http://www.b3.com.br/pt_br/noticias/investidores.htm. Acesso em: 22 mar. 2021.

CANTO, L. G. **Análise de notícias do mercado financeiro utilizando processamento de linguagem natural e aprendizado de máquina para decisões de *swing trade***. Trabalho de Conclusão de Curso (Especialização) - Escola Politécnica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2020.

CHARU C. AGGARWAL. **Data Mining: The textbook**. v. 53. New York: Springer, 2013.

CHATTERJEE, S. et al. Exploring healthcare/health-product ecommerce satisfaction: A text mining and machine learning application. **Journal of Business Research**, Jan. 2020.

CHEN, L.; SHAN, W.; LIU, P. Identification of concrete aggregates using K-means clustering and level set method. **Structures**, v. 34, p. 2069–2076, Set. 2021.

CHRISTIAN, H.; AGUS, M. P.; SUHARTONO, D. Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency. **ComTech: Computer, Mathematics and Engineering Applications**, v. 7, n. 4, p. 285, 2016.

DE ÁVILA, R. L. F.; SOARES, J. M. U**so de técnicas de pré-processamento textual e algoritmos de comparação como suporte à correção de questões dissertativas:** experimentos, análises e contribuições. Congresso Brasileiro de Informática na Educação. p. 727–736, 2013.

DIOGO, J.; FERREIRA, C. Python para Pré-processamento e Extracção de Características a partir de Texto Português. Dissetação (Mestrado em Engenharia Informática) - Faculdade de Ciências e Tecnologia p. 1–88, 2019.

DUARTE, M. A. et al. Estudo Da Estatística De análise Temporal Com Inteligência Artificial Para Aplicação Em Robos De Investimento No Auxílio Dos Investidores Para Otimização De Ganhos Na Bolsa De Valores De São Paulo. **Revista Computação Aplicada - UNG-Ser Aplicada - UNG-Ser**, v. 7, n. 1, p. 12, 2019.

FAHIM, A. K and starting means for k-means algorithm. **Journal of Computational Science**, v. 55, p. 101445, Set. 2021.

FELDMAN, R.; SANGER, J. The text mining handbook: Advanced approaches in analyzing unstructured data. **Cambrge University Press**, 2007.

FERNANDES, M. V. **O fenômeno blockcahin na perspectiva da estratégia tecnológica:** uma análise de conteúdo por meio da descoberta de conhecimento em texto. Dissertação (Mestrado em Gestão de Negócios) - Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Gestão de Negócios, Porto Alegre, 2018.

FERREIRA, A. B. DE H. **Mini dicionário aurélio básico da língua portuguesa**. 4. ed. 2009.

GALDI, F. C.; LOPES, A. B. Relação de longo prazo e causalidade entre o lucro contábil e o preço das ações: evidências do mercado latino-americano. **Revista de Administração - RAUSP**, v. 43, n. 2, p. 186–201, 2008.

GLOBO. **Notícia da morte de Michael Jackson derruba Google e Twitter**. G1. 26 jun. 2019. Disponível em: https://g1.globo.com/Noticias/Musica/MUL1208628-7085,00-NOTICIA+DA+MORTE+DE+MICHAEL+JACKSON+DERRUBA+ GOOGLE+ E+TWITTER.html. Acesso em: 18 out. 2021.

HADDI, E.; LIU, X.; SHI, Y. The role of text pre-processing in sentiment analysis. **Procedia Computer Science**, v. 17, p. 26–32, 2013.

INFOMONEY. **Petrobras**. Disponível em: https://www.infomoney.com.br/cotacoes/petrobras-petr4/historico/. Acesso em: 19 out. 2021.

JIANG, X. L.; ZHANG, K.; YIN, J. F. Randomized block Kaczmarz methods with k-means clustering for solving large linear systems. **Journal of Computational and Applied Mathematics**, v. 403, n. 11601323, p. 113828, 2022.

JULIBONI, Márcio. **Volume movimentado pela B3 salta 71%, em 2020, e quase empata com o PIB pela primeira vez**. Disponível em: https://www.moneytimes.com.br/volume-movimentado-pela-b3-salta-71-em-2020-e-quase-empata-com-o-pib-pela-primeira-vez/. Acesso em: 11 jan. 2021.

KATAYAMA, D.; TSUDA, K. A method of using news sentiment for stock investment strategy. **Procedia Computer Science**, v. 176, p. 1971–1980, 2020.

KINYUA, J. D. et al. An analysis of the impact of President Trump's tweets on the DJIA and S&P 500 using machine learning and sentiment analysis. **Journal of Behavioral and Experimental Finance**, v. 29, 2021.

KONONOVA, O. et al. Opportunities and challenges of text mining in aterials research. **iScience**, v. 24, n. 3, p. 102155, 2021.

KRAAIJ, W. Porter's stemming algorithm for Dutch. **Informatiewetenschap**, p. 167–180, 1994.

LIU, R. et al. Predicting shareholder litigation on insider trading from financial text: An interpretable deep learning approach. **Information and Management**, v. 57, n. 8, 2020.

LOPER, E.; BIRD, S. NLTK: The Natural Language Toolkit. 2002.

MARIZ, B. J. L. **Avaliação de impacto do "Joesley Day" sobre o risco Brasil, e retorno e volatilidade do Ibovespa utilizando a metodologia artificial counterfactual (ArCo)**. Fundação Getulio Vargas- Escola de Pós-Graduação em economia mestrado em finanças e economia empresarial, Rio de Janeiro, 2020.

MARTINS, C. A.; MONARD, M. C.; MATSUBARA, E. T. PreTexT: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words. **Icmc-Usp**. Ago. 2003.

MERLINI, D.; ROSSINI, M. Text categorization with WEKA: A survey. **Machine Learning with Applications**, v. 4, p. 100033, Nov. 2021.

MISHRA, S. P. et al. Principal components analysis. **Atencion primaria / Sociedad Española de Medicina de Familia y Comunitaria**, v. 12, n. 6, p. 333–338, 2017.

MOHAMED, A. A. An effective dimension reduction algorithm for clustering Arabic text. **Egyptian Informatics Journal**, v. 21, n. 1, p. 1–5, 2020.

MORAIS, E. A. M. **Mineração de Textos**. Instituto de Informática da Universidade Federal de Goiás. Goias, Dez. 2007.

MPF. **Operação lava jato**. Disponível em: http://www.mpf.mp.br/grandes-casos/lava-jato. Acesso em: 27 abr. 2021.

MUNKOVÁ, D.; MUNK, M.; VOZÁR, M. Data pre-processing evaluation for text mining: Transaction/sequence model. **Procedia Computer Science**, v. 18, p. 1198–1207, 2013.

NAGHIZADEH, A.; METAXAS, D. N. Condensed silhouette: An optimized filtering process for cluster selection in K-means. **Procedia Computer Science**, v. 176, p. 205–214, 2020.

OLSON, R. S. **Python Machine Learning**. Birmingham: Packt, 2015.

PARAMANIK, R. N.; SINGHAL, V. Sentiment analysis of Indian stock market volatility. **Procedia Computer Science**, v. 176, p. 330–338, 2020.

PATEL, J. et al. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. **Expert Systems with Applications**, v. 42, n. 1, p. 259–268, 2015.

PORTER, M. An algorithm for suffix stripping. **Program 14**, p. 130–137, 1980.

RAMINELLI, D. G. DE T. L.; SANTOS, B. DOS S. Aplicação de Técnicas de Mineração de Dados e Aprendizagem de Máquina no Mercado de Ações : Uma Revisão Sistemática. **Congresso Brasileiro de Engenharia de Produção**, v. 9, Dez. 2019.

RIBEIRO, E. R. **Impacto de técnicas de pré-processamento de texto na detecção de intenção e extração de parâmetros em sistemas de diálogo orientados a tarefa.** Universidade Federal do Amazonas, Programa de Pós-Graduação em Informática, Amazonas, 2020.

RIZÉRIO, L. **Juntas, Cia. Hering e Arezzo ganham R$ 1,4 bilhão de valor de mercado em um dia, mesmo após fusão ser negada.** Disponível em: https://www.infomoney.com.br/mercados/hgtx3-arzz3-disparada-acoes-cia-hering-apesar-de-proposta-fusao-arezzo-rejeitada-credit-suisse/. Acesso em: 27 abr. 2021.

SEONG, N.; NAM, K. Predicting stock movements based on financial news with segmentation. **Expert Systems with Applications**, v. 164, 2021.

SHRIFAN, N. H. M. M.; AKBAR, M. F.; ISA, N. A. M. An adaptive outlier removal aided k-means clustering algorithm. **Journal of King Saud University - Computer and Information Sciences**, 2021.

SILVA, G. M.; TESSARO, N. T. Análise de correlação entre indicadores financeiros e variação de preços de ações utilizando mineração de dados. **Caderno Organização Sistêmica**, v. 2, n. 2, p. 37- 52, 2013

SOARES, M. V. B.; PRATI, R.; MONARD, M. C. PreTexT II : Descrição da Reestruturação da Ferramenta de Pré-Processamento de Textos. **Icmc-Usp**, 2008.

SOUZA, W. B. C. **Mineração de dados aplicada a previsão de preços de ações utilizando Weka.** Dissetação - Escola de Ciências Exatas e da Computação,

Pontifícia Universidade Católica de Goiás, Goiânia, 2021.

SUBBALAKSHMI, C. et al. A method to find optimum number of clusters based on fuzzy silhouette on dynamic data set. **Procedia Computer Science**, v. 46, p. 346–353, 2015.

TRSTENJAK, B.; MIKAC, S.; DONKO, D. KNN with TF-IDF based framework for text categorization. **Procedia Engineering**, v. 69, p. 1356–1364, 2014.

TWITTER, Inc. **Como usar o Twitter**. Disponível em: https://help.twitter.com/pt/using-twitter. Acesso em: 18 out. 2021.

TWITTER, Inc. **Limites da Twitter API**. Developer Platform. Disponível em: https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits. Acesso em: 21 out. 2021.

TWITTER, Inc. **Política de privacidade**. 19 ago. 2021. Disponível em: https://twitter.com/pt/privacy. Acesso em: 19 out. 2021.

TWITTER, Inc. **Twitter API**. Developer Platform. Disponível em: https://developer.twitter.com/en/products/twitter-api. Acesso em: 19 out. 2021.

ULFENBORG, B. et al. Multi-assignment clustering: Machine learning from a biological perspective. **Journal of Biotechnology**, v. 326, n. December 2020, p. 1–10, 2021.

VALE. **Samarco Mineração, Barragem de Fundão, Brasil**. Disponível em: http://www.vale.com/PT/aboutvale/transparencia-e-sustentabilidade/Paginas/Principais%20Desafios/Samarco-Minera%C3%A7%C3%A3o-Barragem-de-Fund%C3 %A3o- Brasil.aspx27. Acesso em: abr. 2021.

YADAV, A. et al. Sentiment analysis of financial news using unsupervised approach. **Procedia Computer Science**, v. 167, p. 589–598, 2020.